VBA: Join-Like Geometry Transfer in ArcMap

Contributed by Bert Granberg
03, May. 2010
Last Updated 04, May. 2010

Written to handle NTIA broadband mapping project census block and road segment level submissions, this VBA script is designed to run within ArcMap to handle a 'join-and-calculate' operation where geometry in transferred from one SHAPE column/field to another.

In Utah, we expect many broadband providers (Qwest, Comcast, local telcoms, etc) to submit data defining their service areas by sending in a text/tabular list of census block IDs together with information describing the type and speed of the available service. As with all of the broadband data in our state, we would like to get this into the Version 2 NSGIC Broadband Data Model in file geodatabase format.

Unfortunately, in the standard ArcGIS desktop user interface, we haven't found a way to join the provider's tabular data to the census block and road geometry in an efficient manner. The workflow and code below attempt to do this given an provider text file
- Open text file submission and add or clean up field name headers as needed
- Remove all commas from text file using a careful search and replace. In Utah we are eliminating commas where unnecessary and replacing with pipe (|) character where we want to preserve the delimiter
- Save text file as ArcMap conforming .csv file
- Load text file into relavent empty V2 NSGIC BBDM feature class (referred to as target feature class below)

- Add this target feature class and "source" feature class containing feature id's and geometry already in desired projection/coordinate system
- Add script to ArcMap VBA editor

- Set script parameters (look for '***** SET SET SCRIPT PARAMETERS in script) for layer positions and names of respective key attribute field names
- Run script

Notes:

- This process works especially well when a single provider submits multiple records for a census block (e.g. where the provider uses multiple technologies)

- Since this script is storing each feature that get assigned a geometry, it's also the ideal time to make edits and additions to the attribute values in the target feature class. Some examples are shown, commented out, in the code below. Note that there are three sections that must be uncommented to make this work: variable declaration (dim statements), findfield statements, and value assignments. You'll want/need to customize these to perform the results you're looking to produce.

```
Option Explicit

Public Sub transferGeometryUsingKeyAttributes()

    'AGRC-BG 20100503, Updated 05/04/2010 12:51pm

    'Uses tablesort object and cursors to produce a 'join-and-calculate' operation
    'with ArcMap where the geometry is passed form one feature class to another.
    'Script written for adding geometry to text-based Census Blocks and
    'TIGER Road Segment Broadband service submissions within the V2 NSGIC
    'Broadband Data Model. Prerequisite step is loading all the tabular data into a
    'feature class (SHAPE geometry column will exist with empty geometry values)

    'Src indicates the layer containing the geometry to be carried over
    'Tar indicates the layer of the same geometry type as src layer.
    '    The tar layer is the layer with empty (or to be overwritten)
    '    SHAPE geometry column that you wish to populate from the Src layer

    Debug.Print "Start: " & Now
```

```
Dim pMxDoc As IMxDocument
Dim pMap As IMap

Dim pSrcFeatureLayer As IFeatureLayer
Dim pSrcFeatureClass As IFeatureClass
Dim pSrcFeature As IFeature
Dim pSrcTableSort As ITableSort
Dim pSrcFeatureCursor As IFeatureCursor
Dim pSrcQueryFilter As IQueryFilter
Dim srcIDFieldName As String
Dim srcID As String
Dim srcIDFieldPosition As Long

Dim pTarFeatureLayer As IFeatureLayer
Dim pTarFeatureClass As IFeatureClass
Dim pTarFeature As IFeature
Dim pTarTableSort As ITableSort
Dim pTarFeatureCursor As IFeatureCursor
Dim pTarQueryFilter As IQueryFilter
Dim tarID As String
Dim tarIDFieldName As String
Dim tarIDFieldPosition As Long

Dim tarCount, count As Long

'optional, for use in parsing blockid, adding census block year, and state identifier
Dim tarSTFieldPosition, tarCOFieldPosition, tarTractFieldPosition, _
   tarBlockFieldPosition, tarVintageFieldPosition, tarGUIDFieldPosition As Long
Dim pUID As UID
Set pUID = New esriSystem.UID

Set pMxDoc = ThisDocument 'This is a module-level variable!
Set pMap = pMxDoc.FocusMap

'***** SET SET SCRIPT PARAMETERS
Set pSrcFeatureLayer = pMap.Layer(0) 'Position of layer, in arcmap TOC, containing geometry to be transferred
Set pTarFeatureLayer = pMap.Layer(1) 'Position of layer, in arcmap TOC, geometry will be transferred to
srcIDFieldName = "TLID"          'Name of Field containing the key field in src feature class
tarIDFieldName = "Identifier"          'Name of Field containing the key field in src feature class
'***** END SET SCRIPT PARAMETERS

If pSrcFeatureLayer Is Nothing Or pTarFeatureLayer Is Nothing Then
   MsgBox "Data not processed. Check source and target position indexes"
End If

Set pSrcFeatureClass = pSrcFeatureLayer.FeatureClass
Set pTarFeatureClass = pTarFeatureLayer.FeatureClass

If pSrcFeatureClass.FeatureType <> pTarFeatureClass.FeatureType Then
   MsgBox "Data not processed. Check source and target position indexes. Geometry type must match"
End If

srcIDFieldPosition = pSrcFeatureClass.FindField(srcIDFieldName)
tarIDFieldPosition = pTarFeatureClass.FindField(tarIDFieldName)

If srcIDFieldPosition <> -1 And tarIDFieldPosition <> -1 Then

   Set pSrcFeatureClass = pSrcFeatureLayer.FeatureClass
   Set pSrcQueryFilter = New QueryFilter
   pSrcQueryFilter.WhereClause = "not " & srcIDFieldName & " is null"
   Set pSrcTableSort = New TableSort
   With pSrcTableSort
      Set .Table = pSrcFeatureClass
      Set .QueryFilter = pSrcQueryFilter
      .Fields = srcIDFieldName
```

```
    .Ascending(srcIDFieldName) = True
    .Sort Nothing
End With
Set pSrcFeatureCursor = pSrcTableSort.Rows
Debug.Print "Source Features Sorted: " & Now
Set pSrcFeature = pSrcFeatureCursor.NextFeature


Set pTarQueryFilter = New QueryFilter
pTarQueryFilter.WhereClause = "not " & tarIDFieldName & " is null"
Set pTarTableSort = New TableSort
With pTarTableSort
    Set .Table = pTarFeatureClass
    Set .QueryFilter = pTarQueryFilter
    .Fields = tarIDFieldName
    .Ascending(tarIDFieldName) = True
    .Sort Nothing
End With
Set pTarFeatureCursor = pTarTableSort.Rows
Debug.Print "Target Features Sorted: " & Now
Set pTarFeature = pTarFeatureCursor.NextFeature

'optional, for use in parsing blockid, adding census block year, and state identifier
'tarSTFieldPosition = pTarFeatureClass.FindField("StateFIPS")
'tarCOFieldPosition = pTarFeatureClass.FindField("CountyFIPS")
'tarTractFieldPosition = pTarFeatureClass.FindField("Tract")
'tarBlockFieldPosition = pTarFeatureClass.FindField("Block")
'tarVintageFieldPosition = pTarFeatureClass.FindField("CBYear")
'tarGUIDFieldPosition = pTarFeatureClass.FindField("Identifier")

tarCount = pTarFeatureClass.FeatureCount(pTarQueryFilter)

Do Until pTarFeature Is Nothing

    If Not pSrcFeature Is Nothing Then

        srcID = pSrcFeature.Value(srcIDFieldPosition)
        tarID = pTarFeature.Value(tarIDFieldPosition)

        If srcID = tarID Then

            count = count + 1
            Debug.Print count
            'transfer geometry to target feature and increment just target feature
            Set pTarFeature.Shape = pSrcFeature.Shape

            'optional, for use in parsing blockid, adding census block year, and state identifier
            'pTarFeature.Value(tarSTFieldPosition) = Left(tarID, 2)
            'pTarFeature.Value(tarCOFieldPosition) = Mid(tarID, 3, 3)
            'pTarFeature.Value(tarTractFieldPosition) = Mid(tarID, 7, 6)
            'pTarFeature.Value(tarBlockFieldPosition) = Right(tarID, 4)
            'pTarFeature.Value(tarVintageFieldPosition) = "2009"
            'pUID.Generate
            'pTarFeature.Value(tarGUIDFieldPosition) = pUID.Value

            pTarFeature.Store
            If count / 500 = CLng(count / 500) Then
                Debug.Print ""
                Debug.Print " --> progress: " & count & " of (" & tarCount & ")"
            End If

            Set pTarFeature = pTarFeatureCursor.NextFeature

        ElseIf CLng(srcID) > CLng(tarID) Then
            'increment target feature
```

```
            Set pTarFeature = pTarFeatureCursor.NextFeature
          ElseIf CLng(srcID) < CLng(tarID) Then
            'increment source feature
            Set pSrcFeature = pSrcFeatureCursor.NextFeature
          End If
        Else
          'no matching src census block record found, write flag to target record
          'pTarFeature.Value(tarVintageFieldPosition) = "N/A"
          'pTarFeature.Store
          Set pTarFeature = pTarFeatureCursor.NextFeature
        End If
      Loop
      Debug.Print ""
      Debug.Print " --> progress: " & count & " of (" & tarCount & ")"
      Debug.Print ""
      Debug.Print "End: " & Now
    Else
      MsgBox "Data not processed. Check source and target field names (use base name, not alias)"
    End If
End Sub
```